Subject: [railML2] Extension proposal: train numbers in <trainGroup> and a "group of groups"
Posted by Janne Möller on Tue, 13 Oct 2020 12:45:46 GMT
View Forum Message <> Reply to Message

Dear railML-community,

In the Norwegian railway sector, we would like to be able to characterise a train group by a set of train numbers. This will make it possible for us to map the service the trains of this train group belong to when exchanging data between different actors of the sector. That is why we extended the element <trainGroup> by the attributes @nor:trainNumbersFrom and @nor:trainNumbersTo in railML2.4nor.

We also want to group different train groups. The reason for this is that services of the same line might belong to different train groups because of their deviation of characteristics, but a group of groups would allow us to indicate that they belong to the same service. An example would be a train group of the extra rush trains of a line that can for example have a different stopping pattern than the regular trains, and a train group of the regular trains of this line. We can create a "group of groups" of the two mentioned. This group of groups will contain the train numbers, but not a <trainRef> for every train. By introducing an attribute @nor:parentRef we can refer to the "group of groups" from inside a "normal" train group.

Any feedback is highly appreciated.

Best regards,
Janne Möller
Jernbanedirektoratet

Subject: Re: [railML2] Extension proposal: train numbers in <trainGroup> and a "group of groups"
Posted by on Thu, 26 Nov 2020 12:22:06 GMT
View Forum Message <> Reply to Message

Hello Janne,

Thank you for your explanations. I still have some questions for understanding as well as remarks:

I understand the need to group <trainGroup>s hierarchically and I have no objections to a new attribute parentRef.

If I have understood it correctly, you basically want to use two types of <trainGroup>s:
a) "elementary trainGroups": These are used in the usual way, i.e. they contain references to all <train>s associated it.
(b) "top-level train groups": These have no references to <train>s and are only used as a "parent" for several "elementary trainGroups". For

this purpose the new attribute "parentRef" of the "elementary trainGroup" is used.
In this way a "top-level-trainGroup" can indirectly reference all its <train>s via its subordinated "elementary trainGroups".
Have I understood the concept correctly?

If so, I don't really see any need for the attributes trainNumbersFrom and trainNumbersTo. All train numbers involved can be determined directly from the referenced <train>s.

Best regards
Christian Rößiger
--
iRFP e. K. · Institut für Regional- und Fernverkehrsplanung
Hochschulstr. 45, 01069 Dresden
Tel. +49 351 4706819 · Fax. +49 351 4768190 · www.irfp.de
Registergericht: Amtsgericht Dresden, HRA 9347

Subject: Re: [railML2] Extension proposal: train numbers in <trainGroup> and a "group of groups"
Posted by Milan Wölke on Wed, 09 Dec 2020 11:48:46 GMT
View Forum Message <> Reply to Message

Hi Janne,
although in general I do understand that it might be useful to have hierarchical train groups, I dont exactly understand your example here. Sounds to me that it would be better in the described case to have a train be part in multiple groups each focussing on a certain aspect. Depending on the aspect Im interested in I could then evaluate the one group or the other. Can you please explain a bit more on the background?

Best regards, Milan

Subject: Re: [railML2] Extension proposal: train numbers in <trainGroup> and a "group of groups"
Posted by Torben Brand on Thu, 18 Feb 2021 16:43:03 GMT
View Forum Message <> Reply to Message

Dear TT comunity,

After internal discussions we retract the suggestion for the new attributes @trainNumbersFrom and @trainNumbersTo in <trainGroup>.

We would still like to have the possibility to group traingroups hierarchical with the suggested new attribute @parentRef.

Christians Rössigers interpretation is as intended.

To elaborate on Milans question for more background we have different UC's:

Collections: These are groups of trains (<trainGroup>) that are used in the conceptual planning process where there are still different variations of the same train services. These variations are usually smaller changes to a train services slot, so that the different variations of the trains are "overlapping" each other and not meant to operate at the same time. But one is to be chosen in a later planning stage. There is a need to transfer all variations in the same railML file. As there can be a high amount of variations, these are structured in hierarchical groups of groups. Like for instance: different high-level conceptual scenarios containing different alternatives containing different smaller variants of the individual trains. With this information the receiving system can allow to filter on the appropriate trains in the transferred train group hierarchy.
The above-mentioned example is typical for pattern trains as they are used conceptual, they usually need to be grouped in collections.

A further extension on this UC would be, as discussed in the TT telco, to map variations (like rush hour frequency increase) within a <patternTrain>, where the <trainGroup> is the pattern trains grouping into a pattern train with variations and the referenced <patternTrain>s the pattern train variations. The @parentRef is then needed if you would like to group the pattern train (group) with variations in a collection group or other group.

We think it is important to be able to relay the relations between the groups (and secondarily also not to bloating the file size unnecessary) with the use of @parentRef and keeping the hierarchy information contained, instead of flattening the groups by repeating the trains for each individual group.

With kind regards
Torben Brand on behalf of Jernbanedirektoratet

Subject: Re: [railML2] Extension proposal: train numbers in <trainGroup> and a "group of groups"
Posted by Thomas Nygreen on Tue, 23 Feb 2021 04:51:37 GMT
View Forum Message <> Reply to Message

Dear all,

How do we work around that trainGroup requires at least one trainRef?

Best regards,
Thomas
